CMPT 215.3 IN-LAB EXAMINATION
(Group A)
October 30, 2000

Total Marks: 10   **OPEN BOOK and OPEN NOTES**
Time: 90 minutes (3:00pm-4:30pm)

Instructions

Both questions are programming questions. You'll need to design algorithms, write, debug, and test MIPS assembly programs. The marks for each question are as indicated. Allocate your time accordingly.

1. **(5 marks)** Read in from the keyboard a sequence of **non-zero** integers finished by a 0, and print the number of integers that are strictly smaller than (i.e. '<') the first integer, and the number of integers that are strictly greater than (i.e. '>') the first integer. Note that the last 0 does not belong to this sequence of integers. For example, if the user inputs the following eight numbers: 10, 3, 17, 57, 10, 4, 32, 0, the output should be:

> *There are 2 integers smaller than the first integer 10*
> *There are 3 integers greater than the first integer 10*

2. **(5 marks)** Implement function **int *search_delete*(int *data*, int *array_addr*, int *array_len*)** that searches the first occurrence of the integer *data* in an integer array. If the *data* is found, it is deleted from the array by moving the last element of the array into its position, and the index of the *data* is returned. As a result, the length of the array is decreased by 1. If the *data* is not found, −1 is returned instead, and the length of the array remains the same since no removing happens. Note that the index of the array starts at 0. In the function, the *array_addr* is the starting address of the integer array, and the *array_len* is the total number of integers in the array. For example, assume the original array is 7, 24, 15, 3, 15, 10, 9, if you search and delete 15 from the array, the new array will be 7, 24, 9, 3, 15, 10 as the first occurrence of 15 is replaced by the last integer 9.

In your ***main***, you should test the function *search_delete* by first creating an array with seven integers 7, 24, 15, 3, 15, 10, 9, and then calling the function as follows:

> *Index1= search_delete(15, array_addr, array_len_1)*
> *Index2= search_delete(17, array_addr, array_len_2)*
> *Index3= search_delete(10, array_addr, array_len_3)*

The correct value for the array length should be computed based on the return value of the previous function call. After each deleting, print the index returned by the function, and the resulted new array. Note that " " (space) should be used to separate array elements from each other when printing the array.